

Für die folgenden Algorithmen **MinSuche**, **MinSuche2**, **SelectionSort** gilt:

Nach Eingabe einer natürlichen Zahl  $n$  werden den  $n$  Komponenten  $a[0], a[1], \dots, a[n-1]$  einer Liste  $a$  Zufallszahlen zugewiesen.

Die Quelliste und die verarbeitete Liste werden jeweils ausgegeben.

### Algorithmus **MinSuche**

Der Algorithmus **MinSuche** bestimmt das kleinste Element der Liste  $a$  und weist es der Komponente  $a[0]$  zu.

Quelltext in Python:

```
#Eingabe
n = int(input('Anzahl der Datenelemente = '))

# Erzeugen der Liste a mit
# den n Komponenten a[0], . . . , a[n-1]
a = list(range(1,n+1))

# Zuweisung von Zufallszahlen
from random import randint
for i in range(0,n):
    a[i] = randint(1,1000)

# Ausgabe der Quelliste
print('Quelliste:')
print(a)

# Bestimmung des kleinsten Elements
min = a[0]
for i in range(1,n):
    if a[i] < min:
        min = a[i]
        a[i] = a[0]
        a[0] = min

# Ausgabe
print('verarbeitete Liste:')
print(a)
```

Beispiel für  $n = 10$ :

```
Anzahl der Datenelemente = 10
Quelliste:
[285, 972, 236, 304, 306, 40, 662, 621, 820, 636]
verarbeitete Liste:
[40, 972, 285, 304, 306, 236, 662, 621, 820, 636]
```

### Algorithmus **MinSuche2**

Der Algorithmus **MinSuche2** bestimmt die zwei kleinsten Elemente der Liste a und weist diese den Komponenten a[0] und a[1] zu.

Quelltext in Python:

```

n = int(input('Anzahl der Datenelemente = '))

a = list(range(1,n+1))

from random import randint
for i in range(0,n):
    a[i] = randint(1,1000)

# Ausgabe der Quelliste
print('Quelliste:')
print(a)

# Bestimmung des kleinsten Elements
min = a[0]
for i in range(1,n):
    if a[i] < min:
        min = a[i]
        a[i] = a[0]
    a[0] = min

# Bestimmung des zweitkleinsten Elements
min = a[1]
for i in range(2,n):
    if a[i] < min:
        min = a[i]
        a[i] = a[1]
    a[1] = min

# Ausgabe
print('verarbeitete Liste:')
print(a)

```

Beispiel für n = 10:

```

Anzahl der Datenelemente = 10
Quelliste:
[597, 81, 284, 703, 131, 891, 263, 989, 940, 904]
verarbeitete Liste:
[81, 131, 597, 703, 284, 891, 263, 989, 940, 904]

```

### Algorithmus **SelectionSort**

(„Sortieren durch direkte Auswahl“)

Der Algorithmus **SelectionSort** bestimmt

- das kleinste Element der Liste  $a[0], a[1], \dots, a[n-1]$  und weist dieses der Komponente  $a[0]$  zu,
- das kleinste Element der Liste  $a[1], \dots, a[n-1]$  und weist dieses der Komponente  $a[1]$  zu,
- das kleinste Element der Liste  $a[2], \dots, a[n-1]$  und weist dieses der Komponente  $a[2]$  zu,
- $\dots$
- $\dots$
- das kleinste Element der Liste  $a[n-2], \dots, a[n-1]$  und weist dieses der Komponente  $a[n-2]$  zu.

Auf diese Weise gelingt es, die Komponenten der Liste  $a$  der Größe nach zu sortieren.

Quelltext in Python:

```

n = int(input('Anzahl der Datenelemente = '))

a = list(range(1,n+1))

from random import randint
for i in range(0,n):
    a[i] = randint(1,1000)

# Ausgabe der Quelliste
print('Quelliste:')
print(a)

# Sortieren
for j in range(0,n-1):
    min = a[j]
    for i in range(j+1,n):
        if a[i] < min:
            min = a[i]
            a[i] = a[j]
            a[j] = min

# Ausgabe
print('verarbeitete Liste:')
print(a)

```

Beispiel für  $n = 10$ :

```

Anzahl der Datenelemente = 10
Quelliste:
[237, 833, 72, 84, 128, 599, 16, 67, 906, 75]
verarbeitete Liste:
[16, 67, 72, 75, 84, 128, 237, 599, 833, 906]

```